REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
05-07-2011	Technical Paper	JUL 2011 - AUG 2011

4. TITLE AND SUBTITLE
Network Protection with Multiple Availability Guarantees

5a. CONTRACT NUMBER
FA8720-05-C-0002

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Greg Kuperman, Eytan Modiano, and Aradhana Narula-Tam

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
MIT Lincoln Laboratory
244 Wood Street
Lexington, MA 02420

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFLCMC/PZE
20 Schilling Circle, Bldg 1305
Hanscom AFB, MA 01731

10. SPONSOR/MONITOR'S ACRONYM(S)
AFLCMC/PZE

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
We develop a novel network protection scheme that provides guarantees on the time a flow has full connectivity, and guarantees a quantifiable minimum grade of service during a downtime. In particular, a flow can be below the full demand for at most a maximum fraction of time; and then, it must still support at least a fraction q of the full demand. This is in contrast to current protection schemes that offer either full protection or availability-guarantees with no connectivity during the downtime. We develop algorithms for the single and multiple commodity cases for general networks, and show that significant capacity savings can be achieved as compared to full protection. For example, if a connection is allowed to drop to 50% of its bandwidth for 1 out of every 20 failures, then a 24% reduction in spare capacity can be achieved over traditional full protection schemes. For the case of q = 0, which is the standard protection constraint, an optimal pseudo-polynomial timed algorithm is presented.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF: U			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Zach Sweet
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**	SAR	9	**19b. TELEPHONE NUMBER** *(include area code)*
U	U	U			781-981-5997

MS-59442

Network Protection with Multiple Availability Guarantees

Greg Kuperman
MIT LIDS
Cambridge, MA 02139
gregk@mit.edu

Eytan Modiano
MIT LIDS
Cambridge, MA 02139
modiano@mit.edu

Aradhana Narula-Tam
MIT Lincoln Laboratory
Lexington, MA 02420
arad@ll.mit.edu

***Abstract*—We develop a novel network protection scheme that provides guarantees on the time a flow has full connectivity, and guarantees a quantifiable minimum grade of service during a downtime. In particular, a flow can be below the full demand for at most a maximum fraction of time; and then, it must still support at least a fraction q of the full demand. This is in contrast to current protection schemes that offer either full protection or availability-guarantees with no connectivity during the downtime. We develop algorithms for the single and multiple commodity cases for general networks, and show that significant capacity savings can be achieved as compared to full protection. For example, if a connection is allowed to drop to 50% of its bandwidth for 1 out of every 20 failures, then a 24% reduction in spare capacity can be achieved over traditional full protection schemes. For the case of $q = 0$, which is the standard protection constraint, an optimal pseudo-polynomial timed algorithm is presented.**

I. Introduction

As data rates continue to rise, a network failure can cause catastrophic service disruptions. To protect against such failures, networks typically use full protection schemes, which usually double the cost of resources needed to route a connection. An alternative approach to minimize the impact of a failure is to provide a guarantee on the maximum time a connection can be disrupted. This is known as an "availability guarantee", and it is a bound on the fraction of time or probability that a connection can be disrupted. However, these disruptions (downtimes) may be unacceptably long; thus, many service providers opt for the more resource intensive full protection. In this paper, we propose a novel protection scheme with multiple availability guarantees. In addition to the traditional availability guaranteed protection, which allows a complete disruption of flow during a downtime, we guarantee partial connectivity at all times. Thus, our approach is a hybrid between the traditional availability guaranteed and full protection schemes.

Full protection schemes have been studied extensively [1–7]. The most common scheme in backbone networks today is $1+1$ guaranteed path protection [8], which provides an edge-disjoint backup path for each working path, resulting in 100% service restoration after any single link failure. There has also been a growing body of literature for backup provisioning to meet availability guarantees [9–17]. In all of these, primary and backup flows are allocated such that the connection is disrupted for at most a specified fraction of time or probability. During these down-states, the service is completely disrupted.

This work was supported by NSF grants CNS-0626781 and CNS-0830961, by DTRA grants HDTRA1-07-1-0004 and HDTRA-09-1-005, and by the Department of the Air Force under Air Force contract #FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the United States Government.

In this paper, we consider an alternate form of availability guaranteed protection, where a fraction of a demand is guaranteed during the downtime. In particular, a flow is guaranteed to be at least a fraction q of the full demand at all times, and it falls below its full demand for at most a specified downtime. Our novel approach is a form of providing "partial protection".

The partial protection framework was first developed in [18]. More recently, [19] and [20] developed a "theory" of partial protection for both single and multi-commodity settings such that after *any* single link failure, the flow can drop to the partial protection requirement. In [19, 20], a fraction q of the demand is guaranteed to remain available between the source and destination after any failure, where q is between 0 and 1. When q is equal to 1, the service will have no disruptions after any failure, and when q is 0, there will be no flow between the two nodes during the down state. In this paper, we consider meeting partial protection requirements with availability guarantees; i.e. the flow can drop below its full demand for at most a specified downtime. Similar to [12–16], we assume the probability of simultaneous link failures to be negligible and only consider single-link failures.

The novel contributions of this paper include a framework for Multiple Availability Guaranteed Protection (MAGP) for both the single and multiple commodity settings. In particular, the multiple availability guarantees are maintaining the full demand for at least a guaranteed fraction of time, and a guaranteed partial flow during the downtime. Algorithms are developed for both, with sharing of backup resources possible in the case of multiple commodities. For a single commodity with $q = 0$, which has a single availability guarantee and is similar to previous works, we develop an optimal pseudo-polynomial algorithm. We also demonstrate that for a single commodity with $q > 0$, finding a feasible solution to the multiple availability guaranteed protection problem is strongly NP-hard, meaning that there exists no ϵ-approximation, nor pseudo-polynomial optimal, algorithm.

This paper is outlined as follows. In Section II, the model for MAGP is described. In Section III, MAGP is shown to be NP-Hard, and the minimum-cost solution to MAGP is formulated

as an MILP. In Section IV, an optimal pseudo-polynomial algorithm for $q = 0$ is described, and the case when $q > 0$ is shown to be strongly NP-Hard. In Section V, MAGP is extended to multiple commodities.

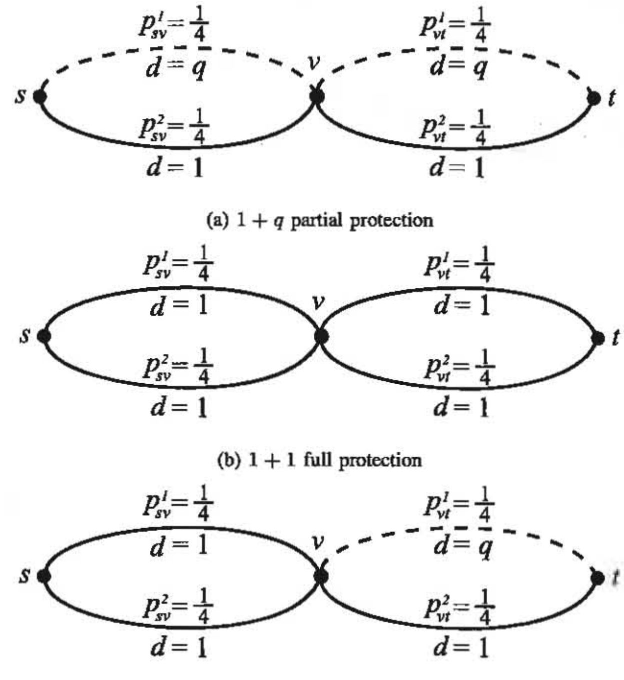
II. Multiple Availability Guaranteed Protection

In this paper, routing strategies are developed and analyzed to minimize the total cost and capacity allocation required to satisfy each demand's guaranteed protection and availability requirements. A demand needs to be routed from its source s to destination t such that upon a link failure, and for at most some specified downtime, at least a fraction q of that demand is guaranteed to remain. To simplify the analysis, we use a "snapshot" model: we consider the network after a failure has occurred. Let p_{ij} be the conditional probability that edge $\{i,j\}$ failed given a network failure has occurred. For simplicity of exposition, instead of a maximum downtime, we consider the Maximum Failure Probability (MFP), denoted as P. The flow can be below the full demand, and at least a fraction q of the demand, with at most probability P. The maximum failure probability can be related to the maximum downtime by accounting for expected time between failures and mean time to repair.

We assume that the graph G, with a set of vertices V, edges E, and probabilities $\mathcal{P}$, is at least two-connected. Since we consider only single link failures, failures are disjoint events, which gives $\sum_{\{i,j\}\in E} p_{ij} = 1$. Similar to previous works (see references in Section I), the primary flow is restricted to a single path. After the failure of a link, a network management algorithm reroutes the traffic along the allocated protection paths. Without loss of generality, for the remainder of this paper we assume unit demands.

Consider the network in Fig. 1, with link failure probabilities and flow allocations as labeled. Suppose we want to route a unit demand from s to t with $P = \frac{1}{4}$ and partial protection requirement q. In [19], a simple partial protection scheme called $1 + q$ protection was developed, which routes the primary demand on one path and the partial protection requirement onto another edge-disjoint path; after any failure in the primary, the partial protection requirement is met. This is shown in Fig. 1a with the solid line carrying the primary flow of 1 and the dotted line carrying the protection flow of q. However, there exists no individual path from s to t that has a failure probability lower than $\frac{1}{2}$. Using the $1 + q$ protection scheme, two edges have an allocation of q, and the user will have a partial flow with probability $\frac{1}{2}$. This failure probabilitiy is greater than the maximum allowed of $\frac{1}{4}$. A naive alternative would be to simply allocate another path for protection, which would be identical to the $1 + 1$ full protection scheme (shown in Fig. 1b); 4 units of capacity are needed and the user will face no downtime, which meets all requirements.

If the primary and backup flows are not restricted to single paths, a more resource efficient allocation can be possible. Consider keeping the primary flow on the same bottom two edges as before, but instead of allocating an end-to-end backup path along the top two edges, we allocate $\frac{1}{2}$ unit flow to protect



(a) $1 + q$ partial protection

(b) $1 + 1$ full protection

(c) MAGP with $P = \frac{1}{4}$

Fig. 1: Comparison of MAGP and traditional protection schemes

against the failure of $\{s,v\}$ and one unit to protect against the failure of $\{v,t\}$ (shown in Fig. 1c). Now, if either of the $\{v,t\}$ edges fail, one unit of flow will still remain from s to t. By fully protecting the primary $\{v,t\}$ edge, there is zero probability that its failure will cause the demand to drop to its partial flow, and the total failure probability of this allocation is $\frac{1}{4}$, which meets the MFP requirement. This routing only needs 3.5 units of capacity, as opposed to the 4 units that full protection requires.

III. Minimum-Cost Multiple Availability Guaranteed Protection

This section investigates minimum-cost allocations for multiple availability guaranteed protection. We assume that each edge $\{i,j\}$ has an associated cost c_{ij}. We start with a negative result regarding the complexity of the multiple availability guaranteed protection problem.

Theorem 1. *Minimum-cost Multiple Availability Guaranteed Protection is NP-Hard.*

Proof: To demonstrate NP-Hardness of MAGP, a reduction from the 1-0 knapsack problem [21] is performed. See Appendix A for proof. ∎

Since the minimum-cost solution to MAGP is NP-hard, we formulate the optimal solution as an MILP. The objective of the MILP is to find a minimum-cost routing to meet a demand's partial protection and availability requirements. In particular, for a connection request between two nodes s and t, the flow can drop to a fraction q of the demand with at most probability P. Again, we are using the snapshot model, and the set of link failure probabilities $\mathcal{P}$ are conditional given a network failure

has occurred. The mixed integer linear program to solve for the optimal routing strategy is given below.

A. *Mixed Integer Linear Program to Meet Multiple Availability Guaranteed Protection*

The following values are given:

- $G = (V, E, C, \mathcal{P})$ is the graph with its set of vertices, edges, costs, and edge failure probabilities
- q is the fraction of the demand between s and t that must be supported on the event of a link failure
- c_{ij} is the cost of link $\{i,j\}$
- p_{ij} is the probability that link $\{i,j\}$ has failed given a network failure has occurred
- P is the maximum probability that the service is below its full demand

The MILP solves for the following variables:

- x_{ij} is primary flow on link $\{i,j\}$, $x_{ij} \in \{0,1\}$
- f_{kl}^{ij} is the protection flow on link $\{i,j\}$ after the failure of link $\{k,l\}$, $f_{kl}^{ij} \geq 0$
- y_{kl}^{ij} is the spare capacity on link $\{i,j\}$ for failure of link $\{k,l\}$, $y_{kl}^{ij} \geq 0$
- z_{kl} is 1 if the failure of link $\{k,l\}$ causes the flow to drop below the primary demand to the partial protection flow, 0 otherwise
- s_{ij} is total spare allocation on link $\{i,j\}$, $s_{ij} \geq 0$

The objective is to:

- Minimize the cost of allocation over all links:

$$\min \sum_{\{i,j\}\in E} c_{ij}(x_{ij} + s_{ij}) \tag{1}$$

Subject to the following constraints:

- Flow conservation constraints for primary flow: route primary traffic to meet demand.

$$\sum_{\{i,j\}\in E} x_{ij} - \sum_{\{j,i\}\in E} x_{ji} = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = t \text{ }, \forall i \in V \\ 0 & \text{o.w.} \end{cases} \tag{2}$$

- Probability constraint: the probability of the set of edges that causes the flow to drop below 1 cannot exceed P. With a single-link failure model, failures are disjoint events and these probabilities become additive.

$$\sum_{\{k,l\}\in E} p_{kl} z_{kl} \leq P \tag{3}$$

- Flow conservation constraints for partial service: if the failure of link $\{k,l\}$ allows the flow to drop below 1, route q from s to t; otherwsie, keep the full primary demand of 1. Let $\mathcal{F}_{kl}$ be the expression $(1 - z_{kl}) + qz_{kl}$.

$$\sum_{\substack{\{i,j\}\in E \\ \{i,j\}\neq\{k,l\}}} f_{kl}^{ij} - \sum_{\substack{\{j,i\}\in E \\ \{j,i\}\neq\{k,l\}}} f_{kl}^{ji} = \begin{cases} \mathcal{F}_{kl} & \text{if } i = s \\ -\mathcal{F}_{kl} & \text{if } i = t \text{ }, \\ 0 & \text{o.w.} \end{cases}$$

$$\forall i \in V, \ \forall \{k,l\} \in E \tag{4}$$

- Capacity allocation: primary and spare capacity assigned on link $\{i,j\}$ meets protection requirements after the failure of link $\{k,l\}$.

$$f_{kl}^{ij} \leq x_{ij} + s_{ij}, \quad \substack{\forall\{i,j\}\in E \\ \forall\{k,l\}\in E} \tag{5}$$

A minimum-cost solution will provide an edge allocation such the flow between s and t can drop to a fraction q of the demand with at most probability P.

B. *Comparison to Full Protection*

Multiple availability guaranteed protection is compared to the $1+1$ full protection scheme via a simulation using the NSFNET topology (Fig. 2) with 100 random unit demands. The protection requirement q is set to $\frac{1}{2}$ for all demands. While we mainly focus in this paper on the case where the primary flow is restricted to a single path, we also consider for this simulation allowing the primary flow to be bifurcated. Bifurcating the flow distributes the risk by lowering the loss of primary flow after any edge failure, which lowers the total allocation needed to meet requirements. This is accomplished by relaxing the integer constraint on the primary flow variables in the MILP.

The availability constraint P is varied from 0 to 0.3 by 0.05 increments. All link costs are set to 1, and the probability of failure for any link is proportional to its length, which is reasonable since a longer fiber will have a higher likely hood of being accidently cut. Routing solutions for MAGP were determined using CPLEX. The shortest pair of disjoint paths were used for the $1+1$ protection [22].

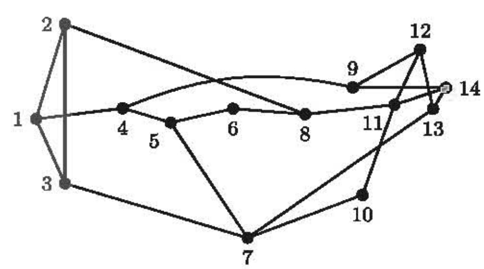


Fig. 2: 14 Node NSFNET backbone network

The average cost to route the demand and protection capacity using the different routing strategies are plotted in Fig. 3 as a function of the maximum failure probability P. The shortest path routing without protection considerations is used as a lower bound for the allocation cost. The cost of providing protection with parameters q and P is the difference between the cost of the respective protection strategies and the shortest path routing.

First, we note that allowing the primary flow to bifurcate allows requirements to met using a lower cost allocation. This is because splitting the primary flow distributes the risk so that upon an edge failure, less primary flow is disrupted. This will then necessitate less protection resources. If the flow is allowed to drop to $\frac{1}{2}$ for 1 out of 20 failures (5% of the time), then a savings of 24% in protection capacity is realized for the case with bifurcation, and 17% without bifurcation. As the flow is allowed to drop more often to its partial protection requirement

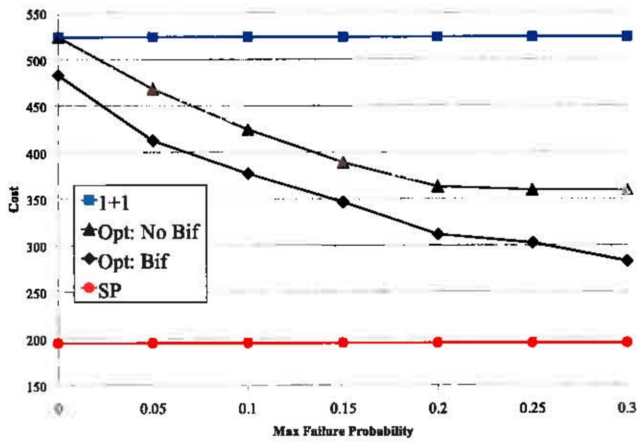


Fig. 3: Capacity cost vs. MFP with $q = \frac{1}{2}$

after a failure, then the savings increase. For 1 out of 10 failures ($P = 0.1$), a savings of 45% and 30% is seen for MAGP with and without bifurcation, respectively. For 1 out of 5 failures ($P = 0.2$), the savings are 65% and 49%. Further increases in P cause only small additional saving; hence, we stopped our simulations at $P = 0.3$.

IV. Optimal Solution and Algorithms

When $q = 0$, we have a single availability guarantee and our problem is to find a path to carry the primary flow, and paths to protect segments of that primary flow, such that the connection is disrupted with probability at most P. This is similar to the problem examined in previous works (see references in Section I). A primary path is identified such that segments of it are protected in a way that after a link failure, the flow drops to 0 with probability at most P. In this section, we consider the single commodity setting, and present a pseudo-polynomial algorithm for finding the minimum-cost solution for protection with availability guarantees when $q = 0$. To the best of our knowledge, this is the first such algorithm. This section is outlined as follows. First, we consider trying to meet availability requirements without the use of protection capacity. Next, we attempt to meet requirements by protecting segments of the primary, and present an optimal pseudo-polynomial algorithm for MAGP when $q = 0$. Then, we examine the case when $q > 0$, and show that a similar problem is strongly NP-Hard, which means that there exists no pseudo-polynomial or ϵ-approximation algorithm. Finally, we present a heuristic for solving the $q > 0$ case.

A. Meeting Availability Requirements Without Spare Allocation

We begin by trying to find the lowest-cost path between s and t such that no protection is required. In other words, finding the lowest-cost path such that the sum of all the failure probabilities in that path are less than P. The MILP for this problem is as follows, with notation and variables the same as in Section III-A,. We assume that all inputs are rational, which is a reasonable assumption for failure probabilities in a network.

$$\min \sum_{\{i,j\}\in E} c_{ij}x_{ij} \tag{6}$$

$$\text{s.t.} \sum_{\{i,j\}\in E} x_{ij} - \sum_{\{j,i\}\in E} x_{ji} = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = t \\ 0 & \text{o.w.} \end{cases}, \forall i \in V \tag{7}$$

$$\sum_{\{i,j\}\in E} p_{ij}x_{ij} \leq P \tag{8}$$

$$x_{ij} \in \{0,1\},\ \forall \{i,j\} \in E \tag{9}$$

We recognize this problem to be the constrained shortest path problem (CSP) [23], which is NP-hard. Instead of the failure probabilities being between 0 and 1, we multiply P and all $p_{ij}, \forall\{i,j\} \in E$, by the smallest factor F that makes all the values integer. A dynamic program exists that finds the minimum-cost solution in pseudo-polynomial time, with a running time of $O(n^2PF)$, where n is the number of nodes in the network; we note that the PF factor is what makes this running time pseudo-polynomial. For simplicity, we return to using the notation P and p_{ij}, but for the remainder of this section they are assumed to be integer. The Bellman equation for this problem was first given in [24] and is presented in Equation 10.

$$f_j(p) = \min\Big(f_j(p-1), \min_{i:p_{ij}\leq p}\big(f_i(w - p_{ij}) + c_{ij}\big)\Big), \quad \forall j \in V - s,\ p = 1, ..., P \tag{10}$$

The recursion finds the minimum-cost constrained path from the source s to any node j, and for every probability of failure $p \leq P$, by taking the minimum-cost of either: (1) an existing path to j with a lower failure probability or (2) a path that is composed of adding edge $\{i, j\}$ to a path from s to some node i that has a total failure probability of at most $(p - p_{ij})$. Once all probabilities p from 1 to P have been recursed, the shortest constrained path from s to t with a maximum failure probability of P can be found by looking up the value $f_t(P)$. This dynamic program can be recognized as a combination of the recursion in the Bellman-Ford shortest path algorithm [23] and the dynamic program to solve the 1-0 knapsack problem [21].

B. Meeting Availability Requirements With Spare Allocation

In general, a path may not exist from source to destination that can meet the availability requirement, nor if one exists, that it is of lowest cost. We next consider the case where certain segments of the primary path will be protected such that the entire end-to-end path meets availability guarantees. A routing that meets these guarantees will be a concatenation of protected and unprotected segments. Consider the routing in Fig. 4, which is an optimal allocation in some network for a unit demand between v_1 to v_6 when $P = 0.2$. The probabilities of link failure are as labeled, and all lines represent a unit flow.

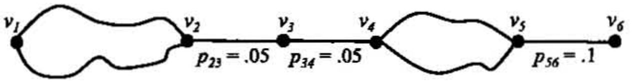


Fig. 4: Routing to meet $P = 0.2$ with $q = 0$ from v_1 to v_6

The primary segments between node pairs (v_2, v_4) and (v_5, v_6) are unprotected, and their total probability of failure must be at most the maximum failure probability of $P = 0.2$. The primary segments between node pairs (v_1, v_2) and (v_4, v_5) are completely protected, and after a failure of either of these primary segments, one unit of flow still remains; they contribute a total failure probability of zero to the routing. We could, in fact, treat each of the protected segments between (v_2, v_4) and (v_5, v_6) as a single edge, with this edge having a probability of zero and a total cost equal to the cost of all the edges used in that segment's primary and protection path. Label the cost and probability of the new edge formed from a protected segment between nodes i and j as $\hat{c}_{ij}$ and $\hat{p}_{ij} = 0$, respectively. We next show that the lowest cost allocation for a segment to have a probability of failure of 0 is the cost of the minimum-cost pair of disjoint paths between i and j.

Lemma 1. *In a network where $p_{kl} > 0$, $\forall \{k, l\} \in E$, the minimum-cost allocation between nodes i and j with a maximum failure probability of 0 is the minimum-cost pair of disjoint paths.*

Proof: Since every edge has a non-zero probability of failure, after any edge failure in the primary path, 1 unit of flow must remain between the source and destination. No edge will have an allocation greater than 1 because the primary flow will have 1 unit, and exactly 1 unit will need to be restored after any primary failure. An equivalent problem is to find the minimum-cost allocation to route 2 units between i and j in a network where every edge has a maximum capacity of 1. This is a minimum-cost flow problem [23], whose solution has integer flows when given integer inputs. Since every edge has a capacity of 1, there will be two distinct edge-disjoint flows of 1 unit each. Clearly, these flows are routed on the minimum-cost pair of disjoint paths. ■

Using Lemma 1, we can transform every possible protected segment in any graph to a single edge with a failure probability of 0 and cost equivalent to the minimum-cost pair of disjoint paths between the two nodes. We denote the cost and probability of the minimum-cost pair of disjoint paths between nodes i and j as $\hat{c}_{ij}$ and $\hat{p}_{ij} = 0$, respectively. We note that the restriction in Lemma 1 of all edges having non-zero probability is not required. Assume some optimal solution includes nodes u and v, and requires a zero failure probability segment between them. Also assume there exists an edge $\{k, l\}$ in the network with $p_{kl} = 0$. Two options that give a zero probability routing between u and v include the pair of disjoint paths between them, or the pair of disjoint paths between u and k, edge $\{k, l\}$, and the pair of disjoint paths between l and v. Any algorithm proposed must be able to capture these possibilities, as well as any others that include possible combinations with zero failure probability edges.

Our proposed algorithm is as follows. First find the minimum-cost pair of disjoint paths between each pair of nodes; there are $O(n^2)$ such pairs. Augment the original graph with an edge between every node pair (i, j) that has cost $\hat{c}_{ij}$ and failure probability $\hat{p}_{ij} = 0$. Finally, run the constrained shortest path algorithm on the transformed graph. We call this algorithm the Segment Protected Availability Guaranteed Algorithm (SPAG).

Theorem 2. *SPAG will return a minimum-cost routing, and has a running time of $O(n^4 log(n) + n^2 P)$.*

Proof: An optimal solution will consist of protected and unprotected segments. With the above graph augmentation, running the constrained shortest path dynamic program will then find the minimum-cost solution. For the running time, the $O(n^4 log(n))$ component comes from $O(n^2)$ iterations of the shortest pair of disjoint paths algorithm, which takes $O(n^2 log(n))$ time per iteration [22]. The recursion for the constrained shortest path problem runs in $O(n^2 P)$ time. ■

C. Meeting Availability Requirements with $q > 0$

Next we consider $q > 0$: after an edge failure in the primary path, the flow either remains at one or, with at most a probability of P, drops to q. An optimal allocation will consist of alternating fully-protected and q-protected segments; a sample solution is shown in Fig. 5 with the dotted line being the q flow. It was shown in Lemma 1 that a fully-protected segment will be the minimum-cost pair of disjoint paths. For the $q = 0$ case, an unprotected segment between some pair of nodes i and j was the shortest constrained path, for which a pseudo-polynomial timed algorithm exists.

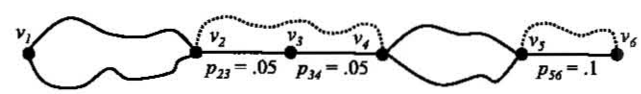


Fig. 5: Routing to meet $P = 0.2$ and $q > 0$ from v_1 to v_6

For a q-protected segment, we need to find the shortest pair of disjoint paths between i and j such that one of them is constrained. We call this problem the Singly Constrained Shortest Pair of Disjoint Paths (SCSPD). There has been work trying to find the shortest pair of disjoint paths such that each path is constrained by the same parameter [25]. The authors found that this doubly constrained problem, while NP-Hard, has an ϵ-approximation algorithm. Their problem is distinct from ours in that we only require one path of the two to be constrained. Clearly, a solution to the doubly constrained problem is a feasible solution to the singly constrained one, but it is not necessarily optimal, and a lack of a solution to the former does not imply the non-existence of a solution to the latter. In fact, we show that when the constraint is relaxed for one of the paths, simply finding a feasible solution to the SCSPD problem becomes *strongly* NP-complete, which means that a solution cannot be ϵ-approximated, nor can there be any pseudo-polynomial algorithm for optimality [21].

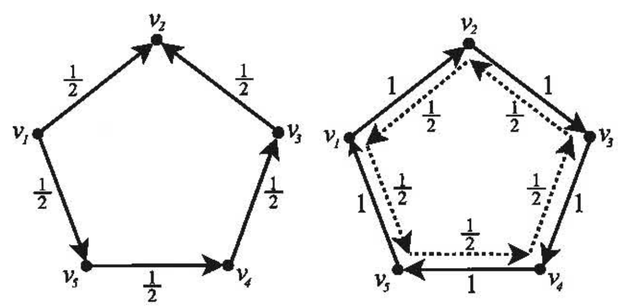
Theorem 3. *Finding a feasible solution to SCSPD is strongly NP-Complete.*

Proof: To demonstrate strong NP-completeness of SCSPD, a reduction from the 3SAT problem [26] is performed. See Appendix B for proof. ■

Because of the strong NP-completeness in finding a feasible solution to SCSPD, the dynamic programming approach used to solve for $q = 0$ will not work when $q > 0$. We conjecture that multiple availability guaranteed protection when $q > 0$ is in fact also strongly NP-hard. Therefore, this necessitates the creation of a heuristic to solve the problem. We propose augmenting the algorithm for the $q = 0$ case: after we solve optimally for $q = 0$, find the shortest disjoint path for the unprotected segments and allocate a flow of q to them. We call this algorithm the Segment Protected Multiple Availability Guaranteed Algorithm (SPMAG).

V. Multi-Commodity Multiple Availability Guaranteed Protection

In this section, we extend multiple availability guaranteed protection framework to the multi-commodity setting. As opposed to the single-commodity version of the problem, when possible, backup resources are shared between demands, lowering the total capacity needed to meet protection and availability requirements. Multi-commodity partial protection without availability guarantees was explored in [20].



(a) v_1 to v_2 without sharing (b) Protection with backup sharing

Fig. 6: No sharing vs. sharing with $q = \frac{1}{2}$ and $P = 1$

To demonstrate how protection sharing can reduce the total capacity needed, consider the example in Fig. 6, with five single hop demands: (v_1, v_2), (v_2, v_3), (v_3, v_4), (v_4, v_5), and (v_5, v_1), each with a partial protection requirement of $q = \frac{1}{2}$ and a maximum failure probability of $P = 1$. The minimum-cost solution for each demand individually is to bifurcate its flow with $\frac{1}{2}$ flow on each direction of the ring. Each demand will require 2.5 units of allocation [19]. The routing for v_1 to v_2 is shown in Fig. 6a, and the routing for the others is not shown but similar. No spare allocation is used and a total of 12.5 units is required to meet primary and protection requirements. Notice that with a single-link failure model, disjoint primary paths will never fail simultaneously; hence, they can share protection resources. Consider routing each primary demand so that each demand is edge-disjoint from the other, as done in Fig. 6b. Since they will never fail simultaneously, spare capacity of $\frac{1}{2}$ units can be allocated onto each edge of the network in the opposite direction. All the demands can share these protection resources since after any single link failure, and all demands will have their protection requirements met. The total capacity allocated to meet partial protection requirements has decreased to 7.5 units: 1 unit of primary flow for each demand with 2.5 units of shared protection capacity.

As demonstrated in Theorm 1, multiple availability guaranteed protection for a single commodity is NP-Hard, so clearly the shared case is also NP-Hard. In addition, it was shown in [2] that the shared path protection problem without availability guarantees is NP-complete when primary and protection flows are each restricted to a single path. Because of this, we formulate a mixed integer linear program to optimally solve the multi-commodity MAGP problem. The MILP is a straightforward combination of the MILP in Section III-A and the linear program presented in [20]; it is not presented in this paper for brevity. Instead, we focus our attention to a heuristic to efficiently solve the multi-commodity MAGP problem with protection sharing.

We consider the dynamic routing model, where demands arrive one-at-a-time to the network, which is similar to protection models for both with and without availability guarantees [2, 10, 14, 20]. A connection arrives at node s to be routed to node t having a demand of d^{st}, a partial protection requirement of q^{st}, a maximum failure probability of P^{st}, and a hold time of t^{st}. Connections are serviced in the order of their arrival, and once a connection is routed, it can no longer be changed. We assume a minimally two-connected graph $G = (V, E, C, P)$.

In [20], an algorithm for shared backup provisioning with partial protection and no availability guarantees was developed using conflict sets. To determine whether protection resources can be shared, we use a conflict set to identify the amount of backup resources that are used on a given edge to protect the failure of another edge [2, 3]. Define the variable h_{ij}^{kl} to be the number of units of capacity used on edge $\{i, j\}$ to protect against the failure of edge $\{k, l\}$. The maximum number of units allocated on edge $\{i, j\}$ to protect against any edge failure is the total spare allocation on $\{i, j\}$. In Fig. 7, two demands with $q^1 = 1$ and $q^2 = \frac{1}{2}$ are routed; for now, assume no availability guarantees. Both demands use edge $\{i, j\}$ for protection with 1 unit being needed after the failure of $\{k, l\}$ and $\frac{1}{2}$ unit being needed after the failure of $\{m, n\}$. The conflict set for this example is $h_{ij}^{kl} = 1$ and $h_{ij}^{mn} = \frac{1}{2}$.

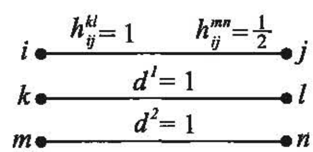


Fig. 7: Example of a conflict set with partial protection

Now, consider a new demand with $q^3 = \frac{1}{2}$. If this demand were to have its primary flow routed on edge $\{k, l\}$ and use $\{i, j\}$ for protection, h_{ij}^{kl} will increase by $\frac{1}{2}$ unit. Since the

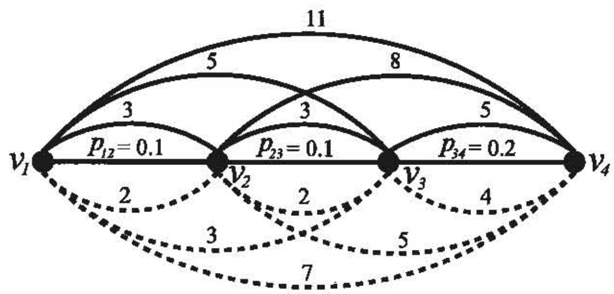
amount of spare allocation on an edge is the maximum capacity needed to protect against any edge failure, the total allocation will increase by $\frac{1}{2}$. Alternatively, if the demand were to use $\{m,n\}$ instead of $\{k,l\}$, p_{ij}^{mn} will increase by $\frac{1}{2}$, and the maximum number of units needed to protect against any edge failure will still only be 1. No additional resources are required for protection on $\{i,j\}$ under this routing scenario.

More generally, we define h_{ij}^{max} to be the maximum allocation needed on $\{i,j\}$ to protect against any edge failure, and we define S to be the edges in the primary path. If $h_{ij}^{kl} < h_{ij}^{max}$, a new demand that uses edge $\{k,l\}$ can share $h_{ij}^{max} - h_{ij}^{kl}$ of protection resources on edge $\{i,j\}$. Denote $b_{ij,kl}^{st}$ as the cost for demand (s,t) to use edge $\{i,j\}$ to protect against the failure of edge $\{k,l\}$. Consider an incoming connection (s,t) with protection requirement q^{st} that has edge $\{k,l\}$ in its primary path and uses edge $\{i,j\}$ for protection. Recall that c_{ij} is the cost of edge $\{i,j\}$ in the original network. If $h_{ij}^{kl}+q^{st} \leq h_{ij}^{max}$, then $b_{ij,kl}^{st} = 0$. Otherwise, $b_{ij,kl}^{st} = c_{ij}[q^{st} - (h_{ij}^{max} - h_{ij}^{kl})]$. We note that this value is never greater than $c_{ij}q^{st}$, because otherwise $h_{ij}^{kl} > h_{ij}^{max}$. The final cost to use edge $\{i,j\}$ for protection is the maximum cost to protect any edge $\{k,l\}$ in the primary using $\{i,j\}$: $\max_{\{k,l\}\in S} b_{ij,kl}^{st}$. Let $B(S,q)$ be the set of network costs associated for some demand with primary path S and protection requirement q.

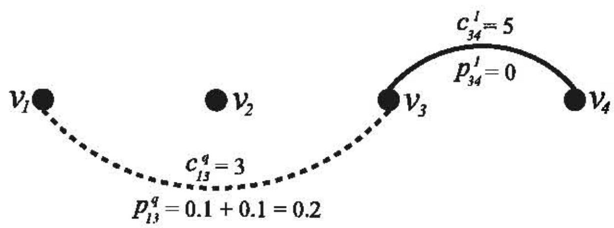
Now, we consider meeting probabilistic availability guarantees. Given some primary path between s and t, certain segments will be fully-protected, and others will be partially protected with a flow of q. For each edge in the primary path, the cost of using edge $\{i,j\}$ for backup is calculated using conflict sets for both 1 or q units of protection. For a primary path with a set of edges S, $B(S,1)$ is the cost of backup edges for for fully protecting any edge, and $B(S,q)$ for partially protecting an edge with a flow of q.

Next, we calculate the cost of protecting each possible segment of the primary path with either full or partial protection; if there are r nodes in the primary path, then there are $\frac{r(r-1)}{2}$ segments contained within that path. We construct a new graph G_S^{st} with two edges between every pair of nodes $\{i,j\}$ in the primary path: one that has cost c_{ij}^1 to fully protect that segment, having failure probability $p_{ij}^1 = 0$, and the other having cost c_{ij}^q to partially protect that segment, having failure probability p_{ij}^q equal to that of the primary path segment $i \rightarrow j$. We then find the shortest constrained path in G_S^{st} from s to t with a maximum failure probability of P^{st}. That path will be the backup, and all partial protection and availability requirements will be met.

Since single-commodity multiple availability guaranteed protection is NP-Hard, and so is jointly optimizing the primary and protection path with backup sharing, we choose a simple strategy of using the shortest path for the primary. S^{st} will be the set of edges in the shortest path between s and t. We then follow the procedure discussed above to find the lowest-cost shared backup to meet protection and availability requirements. This algorithm is called Dynamic Multiple Availability Guaranteed Segment Protection (DMAGSP) and a partially solved example is shown in Fig. 8. The primary path in Fig. 8 is $v_1 - v_2 - v_3 - v_4$, with failure probabilities as labeled. Between every pair of nodes (i,j) of the primary path, we construct two arcs: one that fully protects against a failure in that segment, having probability of failure $p_{ij}^1 = 0$, and one that q-protects that segment, with probability of failure p_{ij}^q equal to the sum of the edges' failure probabilities in that segment. In Fig. 8a, the cost of backup edges for full and partial protection have already been calculated using conflict sets. The arcs above the primary path are the lowest-cost full protection paths for each segment of the primary, and the arcs below the path are the lowest-cost partial protection paths.



(a) All possible protection paths



(b) Final path chosen for backup

Fig. 8: Example of algorithm with $P = 0.2$

The protection paths found by the algorithm, without the primary edges, form the new graph G_S^{st}. Next, we run the constrained shortest path algorithm on G_S^{st} with a maximum failure probability of P, which returns the final backup path. The backup path for this example with $P = 0.2$ (shown in Fig. 8b) meets all protection and availability requirements when combined with the primary path found previously.

We ran a simulation on the NSFNET, similar to that of Section III-B, with demands arriving dynamically and serviced one-at-a-time in the order of their arrival. The protection requirement q for each demand is a truncated normal distribution with mean of $q = \frac{1}{2}$ and standard deviation $\sigma = \frac{1}{2}$. The maximum failure probability P is a truncated normal distributed with a standard deviation $\sigma = 0.025$; the mean of P is varied between 0 and 0.2. We compare optimal multiple availability guaranteed protection with and without sharing, DMAGSP, and 1+1 protection with sharing.

The peak costs to route the demand and protection capacity are plotted in Fig. 9 as a function of the expected value of P. Again, the shortest path routing without protection considerations is used as a lower bound for the allocation cost. Dynamically routed multiple availability guaranteed protection with backup sharing achieves an average reduction in excess

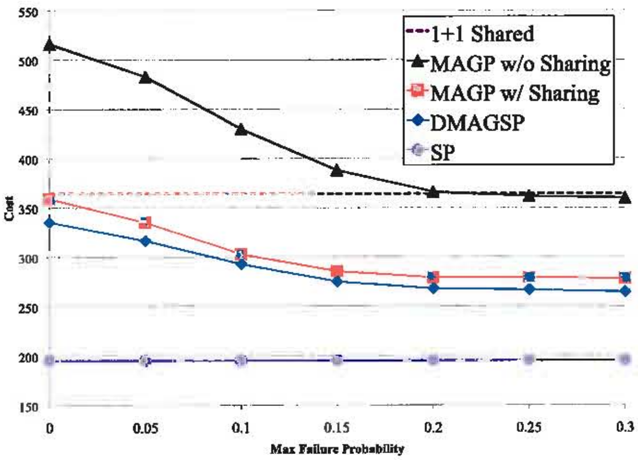


Fig. 9: Peak capacity cost vs. MFP with $q = \frac{1}{2}$

resources of 42% over 1+1 protection with backup sharing, and 51% over MAGP without backup sharing. The most noticeable result is that the algorithm in fact performs *better* than the greedy optimal solution with dynamic arrivals. This can be explained by observing that the algorithm takes the simple strategy of the shortest path as the primary for each connection, as opposed to the jointly optimized primary and backup route, which may take a longer primary path to take advantage of backup sharing. The longer path makes it potentially more difficult for future demands to find disjoint primary routes, lowering their ability to share protection resources. A similar result was also observed in [20] for partial protection without availability guarantees.

VI. Conclusion

In this paper, we introduce the multiple availability guaranteed protection problem (MAGP). We demonstrated the problem to be NP-hard and developed an MILP to find the minimum-cost solution. If the demand is allowed to drop to 50% of its flow for only 1 out of every 20 failures, a 24% reduction in excess resources can be realized over the traditional 1+1 full protection scheme. Next, we presented the first optimal pseudo-polynomial timed algorithm for $q = 0$. For $q > 0$, we showed that finding a feasible solution is strongly NP-complete, and we developed a time-efficient heuristic (MAGSP) to find a solution. We then extended MAGP to the multi-commodity setting, where backup resource sharing is utilized to lower the total capacity needed to meet protection and availability requirements. We developed an algorithm (DMAGSP) that actually performs better than the "optimal" MAGP routing for dynamic arrivals, which jointly optimizes the primary and backup paths for each incoming demand.

Appendix

A. Proof of NP-Hardness for Multiple Availability Guaranteed Protection

To demonstrate NP-Hardness of multiple availability guaranteed protection, the 1-0 knapsack problem [21] will be reduced to MAGP. The knapsack problem is find the maximum value subset of k items, with the i^{th} item having cost c_i and weight p_i, such that the maximum weight P of the knapsack is not exceeded. Consider the network shown in Fig. 10 with link costs and probabilities denoted by c_i and p_i respectively. We wish to find a minimum-cost routing for a unit demand from s to t with a maximum failure probability P and partial protection requirement $q = 0$. After any link failure, the network will either maintain its full flow of 1 unit, or have no flow with a probability of at most P.

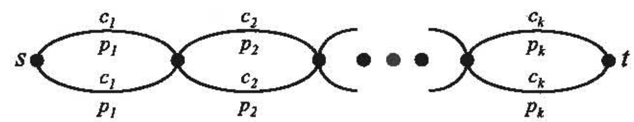


Fig. 10: Sample network for MAGP NP-Hardness proof

There are k distinct link groups, where each of the two links in any group have the same probability of failure and cost. Primary flow has to be allocated onto at least one of these links, otherwise the primary demand cannot be met. If the network maintains full connectivity after a primary failure in the k^{th} link group, then each link in that group will have an allocation of 1 unit. If there is no flow after a link failure, then only one link has an allocation of 1, and the other 0. So, every link group has at least one link with a flow of 1, which is a fixed cost regardless of protection allocation.

To find the lowest cost protection allocation to meet availability guarantees, we find the lowest cost combination of the remaining links after the primary flow is allocated such that the sum of the failure probabilities for the links that have no allocation are less than P: min $\sum_{i=1}^{k} c_i(1 - z_i)$, subject to $\sum_{i=1}^{k} p_i z_i \leq P$, with $z_i \in \{0, 1\}$ $\forall i \in 1, ..., k$. The objective can be rewritten to maximize the cost of the links that do not have allocation: min $\sum_{i=1}^{k} c_i(1 - z_i) =$ max $\sum_{i=1}^{k} c_i z_i$. We now recognize this to be the NP-hard 1-0 knapsack problem with a maximum weight of P, and cost and weight of the i^{th} item being the cost and probability, respectively, of each pair of links in the i^{th} link group. If there existed a polynomial time solution to MAGP, then there would exist one for the 1-0 knapsack problem. Therefore, MAGP is at least as hard as the 1-0 knapsack problem.

B. Proof of Strong NP-Completeness for Singly Constrained Shortest Pair of Disjoint Paths

To prove that finding a feasible solution to Singly Constrained Shortest Pair of Disjoint Paths is strongly NP-complete, we borrow a reduction that demonstrates the NP-completeness of a different, but similar, problem [26] and adapt it to the SCSPD problem. The authors of [26] attempt to find the "min-min" disjoint pair of paths, which is defined as the minimum-cost pair of disjoint paths that contains, over all sets of possible disjoint paths, the minimum-cost shorter path. To demonstrate this problem has no approximation algorithm (and is therefore strongly NP-complete) they construct a mapping of the $3SAT$ problem to a graph where a solution to their problem will simultaneously solve the $3SAT$ problem. A solution to $3SAT$

determines if there exists a 1/0 assignment to the variables that will make a specific boolean expression true [21].

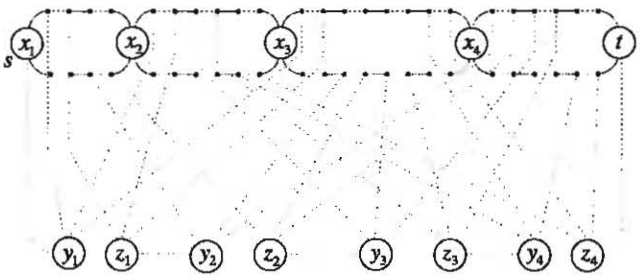


Fig. 11: Sample network to solve an instance of $3SAT$ from [26]

The graph in Fig. 11 is a sample network corresponding to the instance of the $3SAT$ problem of $(x_1 \vee x_2 \vee x_3) \wedge (\hat{x}_1 \vee \hat{x}_3 \vee x_4) \wedge (x_2 \vee \hat{x}_3 \vee x_4) \wedge (\hat{x}_2 \vee x_3 \vee \hat{x}_4)$ [26]. With minimum re-explaination of their reduction (see their work for more details), a generalized version of their result is: if two disjoint paths can be found between s and t such that one of them uses only the dotted lines, then that solution is also a solution to the $3SAT$ problem (see [26] for the proof). They demonstrate that one of the two disjoint paths will have to pass through nodes x_1, x_2, x_3, and x_4. If it passes through the top lobe leaving node x_i, then x_i is true, and false if it passes through the bottom lobe.

To demonstrate strong NP-completeness, one needs to show the problem remains NP-complete even after the value of all inputs to the system have been bounded by some polynomial [21]. Assume there exists D dotted edges and L solid edges in the $3SAT$ reduced graph. Since we can assign parameters of our choosing to the edges, we assign a cost of 0 for the dotted edges and a cost of 1 to the solid edges. We choose the failure probability of each dotted edge to be $\frac{\alpha}{D}$ and the probability of each solid edge to be $\frac{1-\alpha}{L}$, such that $\alpha \leq \frac{1-\alpha}{L} \rightarrow \alpha \leq \frac{1}{1+L}$. Additionally, we choose a maximum probability of failure P such that $\alpha \leq P < \frac{1-\alpha}{L}$. L and D are polynomial bounded by the number of inputs from the $3SAT$ problem, and α can be chosen to be polynomial bounded; so all inputs to the system are bounded. Since using any solid edge will make that path violate the maximum failure probability P, the only feasible solution to SCSPD on this network is for the constrained path to use only dotted edges. But if such a solution could be found, it would solve the $3SAT$ problem, which is NP-Hard. Therefore, the decision problem of finding if a feasible solution exists to SCSPD is strongly NP-complete.

References

[1] S. Ramamurthy, L. Sahasrabuddhe, and B. Mukherjee, "Survivable WDM Mesh Networks," *Journal of Lightwave Technology*, vol. 21, no. 4, p. 870, 2003.

[2] C. Ou, J. Zhang, H. Zang, L. Sahasrabuddhe, and B. Mukherjee, "New and Improved Approaches for Shared-Path Protection in WDM Mesh Networks," *Journal of Lightwave Technology*, vol. 22, no. 5, 2004.

[3] G. Mohan, S. Murthy, and A. Somani, "Efficient algorithms for routing dependable connections in WDM optical networks," *Networking, IEEE/ACM Transactions on*, vol. 9, no. 5, pp. 553–566, 2002.

[4] B. Mukherjee, "WDM Optical Communication Networks: Progress and challenges," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 10, pp. 1810–1824, 2000.

[5] W. Yao and B. Ramamurthy, "Survivable Traffic Grooming with Path Protection at the Connection Level in WDM Mesh Networks," *Journal of Lightwave Technology*, vol. 23, no. 10, p. 2846, 2005.

[6] H. Wang, E. Modiano, and M. Médard, "Partial path protection for WDM networks: End-to-end recovery using local failure information," in *Computers and Communications, 2002. Proceedings. ISCC 2002. Seventh International Symposium on.* IEEE, 2002, pp. 719–725.

[7] W. Grover, *Mesh-Based Survivable Networks: Options and Strategies for Optical, MPLS, SONET, and ATM Networking*. Prentice Hall, 2004.

[8] A. Saleh and J. Simmons, "Evolution Toward the Next-Generation Core Optical Network," *Journal of Lightwave Technology*, vol. 24, no. 9, p. 3303, 2006.

[9] J. Zhang, K. Zhu, H. Zang, N. Matloff, and B. Mukherjee, "Availability-aware provisioning strategies for differentiated protection services in wavelength-convertible wdm mesh networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 15, no. 5, pp. 1177–1190, 2007.

[10] D. Mello, J. Pelegrini, R. Ribeiro, D. Schupke, and H. Waldman, "Dynamic provisioning of shared-backup path protected connections with guaranteed availability requirements," in *Broadband Networks, 2005. BroadNets.* IEEE, 2005, pp. 1320–1327.

[11] J. Zhang, K. Zhu, H. Zang, and B. Mukherjee, "A new provisioning framework to provide availability-guaranteed service in wdm mesh networks," in *Communications, 2003. ICC'03. IEEE International Conference on*, vol. 2. IEEE, 2003, pp. 1484–1488.

[12] M. Tacca, A. Fumagalli, A. Paradisi, F. Unghváry, K. Gadhiraju, S. Lakshmanan, S. Rossi, A. Sachs, and D. Shah, "Differentiated reliability in optical networks: theoretical and practical results," *Journal of lightwave technology*, vol. 21, no. 11, p. 2576, 2003.

[13] R. Banner and A. Orda, "The power of tuning: A novel approach for the efficient design of survivable networks," *Networking, IEEE/ACM Transactions on*, vol. 15, no. 4, pp. 737–749, 2007.

[14] L. Song, J. Zhang, and B. Mukherjee, "Dynamic provisioning with availability guarantee for differentiated services in survivable mesh networks," *Selected Areas in Communications, IEEE Journal on*, vol. 25, no. 3, pp. 35–43, 2007.

[15] W. Yao and B. Ramamurthy, "Survivable traffic grooming with differentiated end-to-end availability guarantees in wdm mesh networks," in *Local and Metropolitan Area Networks, 2004. LANMAN 2004. The 13th IEEE Workshop on.* IEEE, 2004, pp. 87–90.

[16] J. Zhang, K. Zhu, H. Zang, and B. Mukherjee, "Service provisioning to provide per-connection-based availability guarantee in wdm mesh networks," in *Optical Fiber Communication Conference*. Optical Society of America, 2003.

[17] C. Saradhi, M. Gurusamy, and L. Zhou, "Differentiated qos for survivable wdm optical networks," *Communications Magazine, IEEE*, vol. 42, no. 5, pp. S8–14, 2004.

[18] O. Gerstel and G. Sasaki, "Quality of Protection (QoP): a Quantitative Unifying Paradigm to Protection Service Grades," *Optical Networks Magazine*, vol. 3, no. 3, pp. 40–49, 2002.

[19] G. Kuperman, E. Modiano, and A. Narula-Tam, "Analysis and algorithms for partial protection in mesh networks," in *The 30th IEEE International Conference on Computer Communications (INFOCOM 2011)*, 2011, to be published.

[20] ——, "Partial protection in mesh networks: the multi-commodity case," in *Globecom 2011*, 2011, to be published.

[21] C. Papadimitriou and K. Steiglitz, *Combinatorial optimization: algorithms and complexity*. Dover Pubns, 1998.

[22] J. Suurballe and R. Tarjan, "A quick method for finding shortest pairs of disjoint paths," *Networks*, vol. 14, no. 2, 1984.

[23] R. Ahuja, T. Magnanti, and J. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, New Jersey, 1993.

[24] H. Joksch, "The shortest route problem with constraints," *Journal of Mathematical analysis and applications*, vol. 14, pp. 191–197, 1966.

[25] A. Orda and A. Sprintson, "Efficient algorithms for computing disjoint qos paths," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 1. IEEE, 2003.

[26] D. Xu, Y. Chen, Y. Xiong, C. Qiao, and X. He, "On finding disjoint paths in single and dual link cost networks," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 1. IEEE, 2004.